

Snapshot of the SEI Model for e2e: Notes, LIGO-T050141-00-R

Brian Lantz, August 24, 2005

This document two files which represent a snapshot of the SEI model, as of August 24, 2005. This model was used to generate the performance predictions given to the SEI critical design committee.¹ The model is a state space model created by Brian Lantz, using version 2.7 of the modeling tools Wensheng Hua developed for SEI.

The two files can be found in a zip file in the DCC at LIGO-T050142-00-R

The modeling tools allow the user to experiment with various control laws and sensor blending schemes. However, the state space object here does not allow any modification of the control laws or the blending.

The state-space object here uses a reasonable set of control laws above 1 Hz. The cool poly-phase FIR filters used to give good performance in the blending loops for the first stage horizontal loops are not used here, and therefore the performance below 1 Hz is not to be trusted, because it will amplify the tilt motion much more than it should.

I believe there is a bug in the stage 1 rx and ry performance. These loops are not stable, and I believe the reason is from numerical issues. If I fix this, I will promptly release a new model.

The model includes two mechanical stages (each represented in 6 degrees of freedom), sensors, actuators, and control loops. Stage 1 is supported by the ground (or the support structure on top of HEPI). Stage 2 is supported by stage 1. The optics table, which supports the quad pendulum, is part of stage 2. Since the mechanics are all 6 DOF, the standard order is x, y, z, rx, ry, rz. x and y are horizontal translations. There is no “laser beam” which defines a preferred direction, so x and y are essentially the same. Of course, they are not exactly the same, because of the 3-fold symmetry of the sensors and actuators. So far I have not noticed any impact from this. z is the vertical direction. rx is the rotation about the x axis, ry and rz follow naturally. The system is right-handed. Motions in x, y, z are in meters, and rotations in rx, ry, rz are in radians.

The outputs of the 2 stages are with respect to their cg. The optics table is offset vertically from the cg. I'm not sure what the real offset is. At a guess, I'd say 30 cm. The location of the “cg” for the ground is at the cg of the first stage.

Input forces are in newtons for x, y, and z, and in newton-meters for torque.

The sensors are each 1 DOF. The sensor order is horizontal 1, horizontal 2, horizontal 3, vertical 1, vertical 2, vertical 3. For any given set of sensors, the first horizontal sensor (H1) is **located** approximately in the +x direction from the center of the stage, and is **sensitive** to motion in the +y direction at the sensor location (ie both +y motion at the center, and also rz motion about the center. Some of the sensors are not on the same vertical plane as the center of mass of the stage, so they can also be sensitive to motions in rx). H2 and H3 are located at +120 degrees and +240 degrees from H1. The first vertical sensor (V1) of each set is also located in approximately the +x direction, but is sensitive to vertical motion.

A picture of the simulink diagram representing the model is shown below

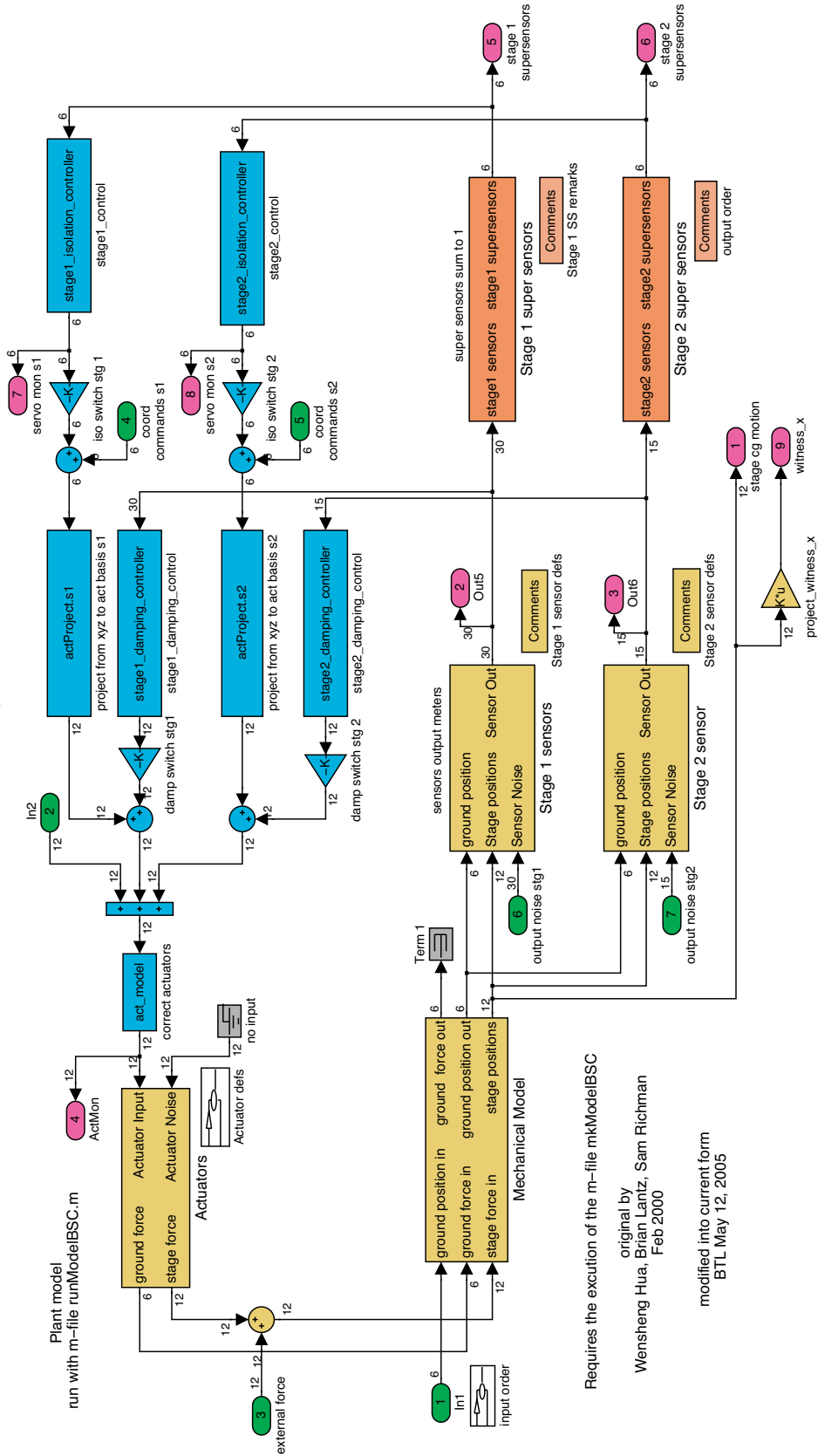
¹ Expectations are:

1) Witness x vector is now `project_witness_x = [zeros(1,6), 1 0 0, 0 .5 .2]`; It used to be `project_witness_x = [zeros(1,6), 1 0 0, 0 .5 0]`; The difference is that now the x witness sensor is explicitly sensitive to rotations about the z axis. Before, the HEPI rz coupling was computed as `.2 * TF(stage2_rz <- gnd_rz) * HEPI_rz`.

2) The out names have been changed, `servomon_nnn` used to be called `actmon_nnn`.

3) The output for the individual center sensors on stage 1 was removed.

Advanced LIGO isolation model – 2 Active stages



Requires the execution of the m-file mkModelBSC

original by
Wensheng Hua, Brian Lantz, Sam Richman
Feb 2000

modified into current form
BTL May 12, 2005

Figure 1: Picture of the Simulink diagram

The displacement sensors are sensitive to the relative motion of the stage at the sensor location with respect to the supporting structure at the same location (in global coordinates). The geophones are sensitive only to the motion of the stage on which they are mounted. The horizontal geophones are also sensitive to both translation and to tilt.

The sensitivity of all the sensors in this model is 1 V/ meter. The tilt sensitivity of the horizontal seismometers is g/w^2 * the translation sensitivity.

The model allows one to couple in sensor noise. The sensor noise is referred to the output noise of the sensor. This allows a sensor to have noise at zero frequency, even if there is no sensitivity there (true for geophones). The transfer function of all the sensors used in this model is 1 V/ meter. So, if your displacement sensors have a white noise of $6e-11$ m/rtHz, you would put $6e-11$ V/rtHz into the output referred sensor noise. If real transfer functions of sensors are implemented in the model, which has been done in the past, the output referred noise will, of course, need to be changed. s

The two files are SEI_main_file.mat and SEI_ref_files.mat. SEI_main_file contains the state space model, and a list of the inputs and outputs. SEI_ref_file contains reference data which describes the various parameters of the model in the snapshot.

```
whos -file SEI_main_file
  Name              Size              Bytes  Class
  SEI_system        1x1              1904364  ss object
  in                1x1              11484   struct array
  note              1x61             122     char array
  out               1x1              12540   struct array
```

These are:

SEI_system the state space model of the system
in a numbered list of all the inputs
out a numbered list of all the outputs
note a note

The **SEI_system** has 94 outputs and 87 inputs. It is sometimes difficult to keep track of which is which, which is why I use the arrays **in** and **out**. This means that to examine the transfer function from the ground motion in the x direction to the output of the x_witness seismometer, you would say

```
TF = SEI_system(out.witness_x, in.gnd_x);
```

instead of

```
TF = SEI_system(94, 1);
```

Note that this does not make use of the cool Matlab feature which allows you to name the inputs and outputs of objects, instead, this is just a big list.

The inputs are:

```
in =
      gnd_x: 1           % support structure translation, meters
      gnd_y: 2
      gnd_z: 3
      gnd_rx: 4         % support structure rotation, radians
      gnd_ry: 5
      gnd_rz: 6
  ActGrdS1H1: 7         % external input to actuator between gnd and stage 1, H1
  ActGrdS1H2: 8
  ActGrdS1H3: 9
  ActGrdS1V1: 10        %      likewise for vertical
  ActGrdS1V2: 11
  ActGrdS1V3: 12
  ActS1S2H1: 13         % external input to actuator between stage 1 and stage 2,
  ActS1S2H2: 14         % for the 3 horizontal acutators
  ActS1S2H3: 15
  ActS1S2V1: 16        % and for the 3 vertical actuators
  ActS1S2V2: 17
```

```

ActS1S2V3: 18
forceS1_x: 19 % pure force on stage 1 cg, newtons
forceS2_x: 25 % pure force on stage 2 cg, newtons
forceS1_y: 20
forceS2_y: 26
forceS1_z: 21
forceS2_z: 27
forceS1_rx: 22 % pure torque about stage 1 cg, newton-meters
forceS2_rx: 28 % and for stage 2
forceS1_ry: 23
forceS2_ry: 29
forceS1_rz: 24
forceS2_rz: 30
stg1_coord_x: 31 % drive in the modal x direction on stage 1
stg1_coord_y: 32 % (pushes against the ground)
stg1_coord_z: 33
stg1_coord_rx: 34
stg1_coord_ry: 35
stg1_coord_rz: 36
stg2_coord_x: 37 % drive in the modal x direction on stage 2
stg2_coord_y: 38 % (pushes against stage 1)
stg2_coord_z: 39
stg2_coord_rx: 40
stg2_coord_ry: 41
stg2_coord_rz: 42
Noise_PSGrdS1H1: 43 % output referred noise of the position sensor
Noise_PSGrdS1H2: 44 % between the gnd and stage 1, H1 - H3
Noise_PSGrdS1H3: 45
Noise_PSGrdS1V1: 46 % and for vertical
Noise_PSGrdS1V2: 47
Noise_PSGrdS1V3: 48
Noise_GeoS1H1: 49 % noise of L4C geophones on stage 1
Noise_GeoS1H2: 50
Noise_GeoS1H3: 51
Noise_GeoS1V1: 52
Noise_GeoS1V2: 53
Noise_GeoS1V3: 54
Noise_StreckX1: 55 % noise for the STS-2
Noise_StreckY1: 56 % note: horizontal servos use Y direction
Noise_StreckZ1: 57 %
Noise_StreckX2: 58
Noise_StreckY2: 59
Noise_StreckZ2: 60
Noise_StreckX3: 61
Noise_StreckY3: 62
Noise_StreckZ3: 63
Noise_GeoS1HT1: 64 % noise for the tilt component of L4C,
Noise_GeoS1HT2: 65 % typically we put all the noise in the
Noise_GeoS1HT3: 66 % horizontal part to avoid confusion
Noise_StreckXT1: 67
Noise_StreckYT1: 68
Noise_StreckXT2: 69
Noise_StreckYT2: 70
Noise_StreckXT3: 71
Noise_StreckYT3: 72
Noise_PSS1S2H1: 73 % noise for position (ie displacement) sensors
Noise_PSS1S2H2: 74 % between stage 1 and stage 2
Noise_PSS1S2H3: 75
Noise_PSS1S2V1: 76
Noise_PSS1S2V2: 77
Noise_PSS1S2V3: 78
Noise_GeoS2H1: 79 % noise of the stage 2 GS-13s
Noise_GeoS2H2: 80
Noise_GeoS2H3: 81
Noise_GeoS2V1: 82
Noise_GeoS2V2: 83
Noise_GeoS2V3: 84
Noise_GeoS2HT1: 85
Noise_GeoS2HT2: 86
Noise_GeoS2HT3: 87

```

The outputs of the model are:

```
out =
    stage1_x: 1           % translation motion of the cg of stage 1, meters
    stage1_y: 2
    stage1_z: 3
    stage1_rx: 4         % rotation motion about the cg of stage 1, radians
    stage1_ry: 5
    stage1_rz: 6
    stage2_x: 7         % translation motion of the cg of stage 2, meters
    stage2_y: 8
    stage2_z: 9
    stage2_rx: 10        % rotation motion about the cg of stage 2, radians
    stage2_ry: 11
    stage2_rz: 12
    SenPSGrdS1H1: 13     % outputs of all the sensors,
    SenPSGrdS1H2: 14     %   output is in "volts" but the sensitivity is
    SenPSGrdS1H3: 15     %   defined as 1 V / meter
    SenPSGrdS1V1: 16
    SenPSGrdS1V2: 17
    SenPSGrdS1V3: 18
    SenGeoS1H1: 19
    SenGeoS1H2: 20
    SenGeoS1H3: 21
    SenGeoS1V1: 22
    SenGeoS1V2: 23
    SenGeoS1V3: 24
    SenStreckX1: 25
    SenStreckY1: 26
    SenStreckZ1: 27
    SenStreckX2: 28
    SenStreckY2: 29
    SenStreckZ2: 30
    SenStreckX3: 31
    SenStreckY3: 32
    SenStreckZ3: 33
    SenGeoS1HT1: 34
    SenGeoS1HT2: 35
    SenGeoS1HT3: 36
    SenStreckXT1: 37
    StreckYT1: 38        % is this a bug?
    SenStreckXT2: 39
    StreckYT2: 40
    SenStreckXT3: 41
    StreckYT3: 42
    SenPSS1S2H1: 43
    SenPSS1S2H2: 44
    SenPSS1S2H3: 45
    SenPSS1S2V1: 46
    SenPSS1S2V2: 47
    SenPSS1S2V3: 48
    SenGeoS2H1: 49
    SenGeoS2H2: 50
    SenGeoS2H3: 51
    SenGeoS2V1: 52
    SenGeoS2V2: 53
    SenGeoS2V3: 54
    SenGeoS2HT1: 55
    SenGeoS2HT2: 56
    SenGeoS2HT3: 57
    ActMon_GrdS1H1: 58   % monitor of the drives going the actuators
    ActMon_GrdS1H2: 59
    ActMon_GrdS1H3: 60
    ActMon_GrdS1V1: 61
    ActMon_GrdS1V2: 62
    ActMon_GrdS1V3: 63
    ActMon_S1S2H1: 64
    ActMon_S1S2H2: 65
    ActMon_S1S2H3: 66
    ActMon_S1S2V1: 67
    ActMon_S1S2V2: 68
    ActMon_S1S2V3: 69
```

```

    SS_stg1_x: 70           % output of the Super Sensor, stage 1, modal x direction
    SS_stg1_y: 71
    SS_stg1_z: 72
    SS_stg1_rx: 73
    SS_stg1_ry: 74
    SS_stg1_rz: 75
    SS_stg2_x: 76
    SS_stg2_y: 77
    SS_stg2_z: 78
    SS_stg2_rx: 79
    SS_stg2_ry: 80
    SS_stg2_rz: 81
    servomon_slx: 82       % loop monitor for the servos, stage 1, x direction
    servomon_sly: 83
    servomon_slz: 84
    servomon_slrx: 85
    servomon_slry: 86
    servomon_slrz: 87
    servomon_s2x: 88
    servomon_s2y: 89
    servomon_s2z: 90
    servomon_s2rx: 91
    servomon_s2ry: 92
    servomon_s2rz: 93
    witness_x: 94         % x motion of the optics table, .5 meters above
                          % and .2 meters to the side of the stage 2 cg. (meters)
    witness_z: 9         % same as the stage 2 z motion.

```

The reference file contains a lot of reference data, which may or may not be useful or interesting. ASI and ref are data structures containing info about the mechanical setup of things, including stage masses. mvReaction is the 12x 12 stiffness matrix for the 2 stages.

```

>> whos -file SEI_ref_file
Name                               Size           Bytes  Class

ASI                                1x1            4660  struct array
mvReaction                          12x12          1152  double array
note                                1x61            122  char array
ref                                  1x1            7160  struct array
stage1_damping_controller           1x1           56766  tf object
stage1_isolation_controller         1x1            8944  zpk object
stage2_damping_controller           1x1           29826  tf object
stage2_isolation_controller         1x1            8136  zpk object

```